

Swift Disc Operating Manual

Author: I. Hunt

Date: 11 September 1987

Software Version: V3

Produced on behalf of: SIXWORD Limited
24 Chatsworth Close
Catisfield
Fareham
Hampshire
PO15 5LS
Tel:(0243) 771689

The information contained herein is the property of Sixword Ltd and is supplied without liability for errors or omissions, and no part may be reproduced, used or disclosed except as authorised by contract or other written permission. The copyright and the foregoing restriction on reproduction, use and disclosure extend to all the media in which this information may be embodied, including magnetic or electronic storage, punch card, paper tape, computer printout, visual display, etc.

Sixword Ltd reserve the right to make any changes to this manual, hardware, firmware and software at any time without prior notice.

In no circumstances will Sixword Ltd be liable for any direct, indirect, incidental or consequential damage or loss of use, data, profit or contracts which may arise from any error, defect or failure of the disc system hardware, software or firmware.

Users should acquaint themselves with the Software Protection Act 1985 (Amendments) and as such all copying of proprietary software should be only under the terms of the licence for the said software.

Sixword Ltd have a policy of continuous improvement.

Index.

Operations from the Button:

	Page
FORMAT	5
SAVE	6
QUIT	7
LOADING	7
CAT	8
ERASE	9
KEEP	9
UNKEEP	10
RENAME	10
MOVE	11
ALTER	12
BACKUP	13

Running from BASIC

CAT	16
LOAD	17
MERGE	18
SAVE	19
OPEN#	20
CLOSE	21
ERASE	21
CLEAR	22
FORMAT	22
SEQUENTIAL FILE ACCESS	24
IN%	25
OUT%	25
%EOF	26
%ERR	26
%LINE	27
%LOWER	27
%DATE	27
PRINTER PORT FORMAT	28

Introduction to Disc Drives.

The Swift Disc System will add the convenience of rapid loading and saving to your ZX Spectrum computer, and provide storage suitable for the games player and the professional user alike. In order to get the best from your system, it is recommended that you read through this manual once before using your system.

The Swift Disc System provides a random access store for programs and data. Unlike the cassette tape, which must be searched through to find and load a program, the Swift Disc System is able to immediately locate and load the required program. A full 48K program can be found and loaded in about 6 seconds.

Setting up your Swift Disc System.

Unpacking the Swift Disc System, you will have found:

- This manual
- The Swift Disc Interface
- A Swift Disc Drive, complete with:
- A ribbon cable, 1.5 metres long (for connecting the Disc Drive to the Interface).

To attach the interface to your computer, you should:

1. Unplug the computer
2. Disconnect the computer from any peripherals
3. Push the interface edge connector firmly into the socket at the back of the computer*
4. Connect the Disc Drive ribbon cable into the connector on the Swift Disc interface box, ensuring that the red stripe on the cable is nearest to the joystick D connector.
Also ensure that all 34 pins in the interface go into the socket.

* Interface 1 and other peripherals.

If Interface 1 and microdrives are required to be run with the Swift Disc, Interface 1 should be connected to the back of your Spectrum first, and the Swift Disc interface should be plugged in behind it. You must bear in mind that the Spectrum will start to misbehave if too many peripherals are connected to it. We would recommend that you do not normally connect a total of more than two or three devices to the Spectrum at the same time. The 128 and +2 will be more susceptible to overloading than the 48K.

Getting Started

Switch on the disc drive and your computer, which will power up normally. Always switch on the drive at the same time or before the computer. The Error Message RECORD NOT FOUND normally results from the computer being switched on first.

Select the diskette that you are going to use. The type of diskette that is required is a 3.5", 135 TPI, double density, double sided disc. At the top of the diskette case is a little plastic write protect shutter that can be pushed across a hole. To enable the computer to store data onto the diskette, the shutter must cover the hole.

Put the diskette into the drive with the label facing upwards, and push it all the way in. To remove the diskette from the drive, press the button on the front of the drive and the diskette will pop out.

The diskettes are fairly robust, but nevertheless, should be handled with care. They should not be stored near magnets, (i.e. the television), and should be protected from contamination when not in use.

Setting your system to work

To begin, activate the Swift Disc Operating System by pressing the interrupt button on the rear of the interface. The interrupt button must only be pressed once when you want to break into a program. Pressing it a second or a third time will reset the internal variables used by the interface and you will have to reset your Spectrum.

Your computer will display:

```
Swift Disc O/S V3.x
```

```
>
```

The '>' is the prompt and replaces the flashing cursor of your computer, when using the Swift Disc Operating System.

All Swift Disc System commands may be abbreviated to a single letter. Throughout the manual the rest of the command is shown (in brackets) for clarity.

Some Special Keys

The BREAK key breaks into any disc operation.

The DELETE key removes characters to the left of the cursor.

Once you have typed a command and entered it, it is then possible to recall it using the EDIT key. The arrow keys move the cursor to the left and the right. Typing any characters will overwrite the current characters, and ENTER will re-enter the command.

If you have forgotten the command, type the first letter, and enter it; the interface will prompt for all the information that it requires, using default values where possible.

How to read the Syntax Guide

Brackets are used in the instructions in this manual, as follows:

() Indicates optional information.

[] Specifies a control key to press, i.e. [SP] = Space bar.

N.B. A DEFAULT value is one that will be used by the interface, if none are specified.

Parameters that are vertically beneath each other indicate an optional substitute.

Operations From the Button

Command Name: FORMAT

Command Syntax:

> F(ormat) [SP] (Name) [SP] (Number) [SP] (Tracks) [ENTER]

Command Options

Name – This is the volume name that you assign to the diskette and will appear in the CAT list. It can have up to 10 characters. No spaces are allowed. The default name is SWIFT.

Number – The Number field represents the maximum number of files that can be stored on the diskette (number of free headers). The default number for this field is 32, but it will accept any number between 14 and 80. The smaller the number, the more rapid the access to the disc files.

Tracks – This is the number of tracks on your disc drive. The default is 80, but 40 can be specified. The disc interface automatically determines the number of sides. This parameter need only be specified if you are NOT using a Sixword supplied disc drive.

Description:

Before a diskette can be used, it must be formatted. Formatting a diskette puts on information that the interface needs to locate your programs. It only needs to be done once.

You will be asked if you really want to continue as formatting will destroy any information on the diskette. Enter Y (or N).

ARE YOU SURE ? > Y(es) [Enter]

The screen will now clear except for the bottom 4 lines. As each of the tracks on the diskette is formatted, the track number is printed on the screen. When all the tracks on the disc have been formatted, the bottom 10 lines of the screen will clear and the prompt will reappear.

Formatting a diskette will take about 1 minute, and can be done at any time. It must be remembered that formatting a diskette destroys everything that was stored on it. When formatting is complete, the prompt (>) will be displayed.

Examples:

> F	This uses all the defaults, as SWIFT, 32,80
> F GAMES	This formats a disc with volume name GAMES
> F FILEDISK 80	This formats the disc with 80 files max.
> F SMALL 32 40	This formats SMALL, 32 headers and 40 tracks

Command Name: SAVE

Command Syntax:

>S(ave) [SP] Filename [SP] (S(screen) [SP]) (Number) [ENTER]
(C(ode) [SP] Start [SP] Len) [ENTER]

Command Options:

Filename - This is a string that can contain up to 10 characters but must not include spaces. A tip is to use filenames that are just long enough to recognise the file, using long names can increase typing errors.

Number - This represents the colour of the border when the file is loaded back. It should be in the range 0-7 as on the Spectrum. Its default value is BLACK.

Screen - This is a Screen save.

Code - This is a code save, from a given START address and a given LENGTH. The decimal numbers defaults may be overridden by hex notation by preceding the number with a #

Description:

Once a program has been loaded into the Spectrum (refer to your Spectrum Introduction Booklet) or Microdrive (if you have Interface 1 and a microdrive connected) you may save it to the diskette. To do this, return to the Swift Disc Operating System by pressing the Interrupt button once.

Note 128 and +2 owners: In the 128K mode, it is possible for a program to use either 1 of 2 screens. If when pressing the Interrupt button, the wrong screen is presented in the top portion of the screen, then pressing the BREAK key will restore you to the correct screen. The BREAK key **MUST** be the first key you press after the Interrupt button, if the wrong screen is presented.

After entering the SAVE command, the Swift Disc Operating System will save the full memory contents on the diskette as an Image file, automatically detecting the 128 mode.

Examples:

>SINVADER	Saves an image called INVADER
>SINVADER 3	Saves the image with a magenta border
>SINVADER S	Saves the screen
>SINVADER C 16384 #1000	Saves 1000h of code starting at 16384

Command Name: QUIT

Command Syntax:

>Q(uit) [ENTER]

Command Options: None

Description:

This will return the computer out of the Disc Operating system to what was running. If you press the button, then you type Q, you will return to BASIC, however if you have loaded a game in the meantime, then you will return to the game.

Example:

>Q

Command Name: LOADING

Command Syntax:

>L(oad) [SP] Filename ([SP] Start ([SP] Length)) [ENTER]

Command Options:

Filename - This is a string of not more than 10 characters, excluding spaces, that tells the interface which program to pick off the diskette.

The Start address and optional length parameters may be specified.

Description:

A file stored on diskette may be loaded at any time. First RESET, NEW, or '*' the computer, (see later in this manual), then press the Interrupt button once to call the Swift Disc Operating System.

Then type the LOAD command as outlined above.

A 48k program will take about 6 seconds to load, after which you will be asked if you want to run the image. Type Y and the program will start to run from the point at which it was saved.

If you answer N, so that you can put in POKES, etc, then the image can be run at any time by Quitting out of the Operating System.

Note that the Y or N does not need to be followed by [ENTER]

Examples:

>LINVADERS	Loads the image INVADERS from the diskette
>LINVADERS #4000	Loads back the image to address 4000H
>LINVADERS #4000 10	Loads back 10 bytes of the Image

Command Name: CAT

Command Syntax:

>C(at) [SP] (Drive_no [SP] B(rief)) [ENTER]
F(ull)

Command Options:

Drive_no – This is the drive number (between 0 and 3) which you require the catalogue of.

Brief – This is the request for a Brief listing, suppressing all information bar the filenames.

Full – This is the Full listing, giving additional information on the filetypes and size.

Description:

Catalogue allows you to see the correct names of the files stored on the diskette. The CATALOGUE will look something like this:

```
CATALOGUE OF: GAMES
INVADERS      192 PI 22-07-86
MYFILE        192 PI 23-07-86
Swift.EXE     192  I 23-07-86
TEMP.BAT      20   B 17-07-86
BASIC.LIS     1   PD 27-12-61
```

```
Total of 5 Files
25 Headers free out of 30
Free Disc Blocks = 1735
```

For a more detailed description of the CAT screen, please see Appendix 1.

The full listing gives record size for record files, start and length for code saves, and autostart line numbers for BASIC programs.

Examples:

```
>C          Gives a catalogue of the default drive.
>C 0 B     Gives a brief catalogue of drive 0.
>C 1 F     Gives a full catalogue of drive 1.
```

Command Name: ERASE

Command Syntax:

>E(rase) [SP] Filename [ENTER]

Command Options:

Filename – This is the name of the file that you wish to be erased.

Description:

Erase permanently erases a file from the diskette unless it has been protected. Once a file has been erased, it cannot be recovered. (See KEEP).

Example:

>E INVADERS

Command Name: KEEP

Command Syntax:

>K(eep) [SP] Filename [ENTER]

Command Options:

Filename – This is the name of the file that you wish to protect.

Description:

The Keep command allows the Protect option to be invoked on a file, making it impossible to Erase the file.

The keep command is used to prevent accidental erasure of a file.

A 'P' is printed in the CAT listing against all files which have been KEEPed.

Note – you can protect the whole diskette with the write protect shutter.

WARNING – Although ERASE will not work on a file that has been KEEPed, FORMAT, which formats a diskette, will delete the KEEPed files along with every other file on the disc.

Example:

>K INVADERS

Command Name: UNKEEP

Command Syntax:

>U(nkeep) [SP] Filename [ENTER]

Command Options:

Filename – This is the name of the file that you wish to remove the protection from.

Description:

This command allows you to remove the keep protection on a file.

Example:

>U INVADERS Unkeeps the file INVADERS

Command Name: RENAME

Command Syntax:

>R(ename) [SP] Oldname [SP] Newname [ENTER]

Command Options:

Oldname – This is the name of the file on the diskette that you wish to change

Newname – This is the newname that you want the file to assume.

Description:

Rename allows you to change the name of any file on the diskette to another filename that is legal, and does not already exist on the diskette.

Example:

>R INVADERS OLDINV Renames the file INVADERS to OLDINV

Command Name: MOVE**Command Syntax:**

>M(ove) [SP] Filename [SP] Copyname (*) [ENTER]

Command Options:

Filename – This is the name of the file that you wish to be copied.

Copyname – This is the name of the newfile on the diskette.

* This allows you to copy a program on to a different diskette using the same drive. The interface will ask you for each diskette in turn, thus you will have to swop diskettes. The default is move onto the same diskette.

Description:

The MOVE command allows copies of a file to be made. If the copy needs to be loaded onto a different diskette than the source, the * option will be specified and diskettes will require to be swapped. If however, the copy is to exist on the same diskette as the source then you should not specify the * option.

If the * is specified, then Spectrum main memory is used. This means that any program that was running in the background will be corrupted.

Examples:

>M INVADERS INVADER1	Make a copy onto the same diskette.
>M INVADERS INVADER1 *	Make a copy onto a different diskette in the same drive and use Spectrum memory.

Command Name: ALTER

Command Syntax:

>A(lter) [SP] Address ([SP] (Data)) [ENTER]

Command Options:

Address – This is a decimal number between 0 and 65535. It can be expressed in decimal (default) or in HEX notation by preceding it with a #.

Data is the value that you want the contents of the address to assume.

Description:

This command allows you to implement POKES etc., that are published in magazines etc.

Examples:

>A 16384 20 Alter the contents of address 16384 to 20

>A #4000 Inspect the contents of address 4000H

Command Name: *

Command Syntax:

>* [ENTER]

Command Options: None

Description:

This command will reset the Spectrum and the disc interface without altering the date. It is useful when backing up many tapes to a diskette

Example:

>*

Command Name: BACKUP

Command Syntax:

>B(ackup) [SP] Source [SP] Destination ([;] Volume label) [ENTER]

Command Options:

Source – This is the number of the drive that you want to use as the source drive, 0 – 3

Destination – This is the number of the destination drive, 0 – 3. If it is the same number as the source drive, then you will be required to swap diskettes.

Volume label – is the name of the diskette that you wish to create

Description:

This command allows you to backup a diskette onto another diskette, either between drives, or onto the same drive. Backing up uses the whole of Spectrum memory to minimise the number of disc swaps

Backup will prompt you for each disc that it requires when it requires it. It is a good idea to write protect the source disc (using the tab) before you start

To backup a disc on a single disc system, you will be required to perform 34 disc changes, so if you are using only IMAGE type files, then it is quicker to load the image into memory and save it again to a different disc. (This only requires 26 disc changes for a full disc).

Examples:

- | | |
|-------------|--|
| >B 0 0;FRED | This is a backup of drive 0 to drive 0 creating a new diskette called FRED (34 disc swaps required) |
| >B 0 1;FRED | This is an interdrive backup from drive 0 to drive 1, also creating a diskette called FRED but with no swapping required |

Other drives.

The Swift Disc INTERFACE is capable of supporting up to 4 drives, numbered drives 0 through to 3. The first drive that you are supplied with is drive 0. The interface initially assumes drive 0 as the default, but this can be modified, as shown below. In general, the method for using another drive is to specify the drive number and a semi colon [;] before the filename, i.e.

```
>L 1;INVADERS
```

This will load Invaders from drive 1, and select drive 1 to be the default.

Useful Hints

All commands can be abbreviated to their first letter. Only single spaces are allowed between instruction parameters. Remember to enter the date when starting a session.

Write protecting a diskette will prevent erasure during a session.

Do not switch on or off with a diskette in the drive.

Do not remove a diskette while the drive is accessing it.

Quit returns you to a normal Spectrum.

Pressing the BREAK key stops any disc operation currently being executed.

Additional Information on the Interface.

Ramtop. The disc interface expects enough stack to be present to operate. No checks are made for ramtop. Therefore always allow 20H words for any access to the disc system.

Fragmentation. The disc interface will do a 'Best Fit' on any files that are saved to the diskette. It is therefore difficult to obtain a fragmented diskette using load and save. However, if you have more than 1 file open at a time and write alternately to all open files, then the data on the diskette will become fragmented. Other than moving the files to another diskette, there is no way that the files can be consolidated.

Streams and Channels

STREAMS and CHANNELS are made use of extensively within your Spectrum. A stream can be imagined as a pipe. Data is forced into the pipe at one end by a device known as a channel, and is read out of the other end of the pipe by another device. The Spectrum can have up to 16 streams at any one time, the first four are already assigned to the system devices.

Stream #0	output data to the lower part of the screen.
Stream #1	input data from the keyboard.
Stream #2	output data to the upper part of the screen.
Stream #3	output data to the printer.

The devices (Channels) are known as "S", "K", "P". Files that are created on the diskette are also treated as devices.

All stream references are preceded by a #

The commands that force data into and out of the streams are PRINT, INPUT, etc., which are extended to PRINT #, and INPUT #.

i.e. PRINT #2;"hello world".

pushes the characters into stream 2 and they are automatically pulled out of stream 2 by the screen, thus displayed.

Running from BASIC

The disc interface will allow you to access the disc from Spectrum BASIC. The BASIC commands that can be used are:—

FORMAT	LOAD
SAVE	ERASE
CAT	OPEN#
CLOSE#	CLEAR
PRINT#	INPUT#
INKEY#	IN
OUT	LET
MERGE	IF

Because the interface is Microdrive compatible, the commands have been tailored to suit the microdrives, enabling microdrives and disc drives to be used together.

All the command keywords that are issued to the disc drive under BASIC are suffixed with the % character, (symbol shift on 5), which tells the Spectrum that the command is for the disc drive.

i.e. OPEN #% or CAT %

NOTE:

() = something that can be included optionally.

< > = a value of some type.

parameters that are vertically beneath each other indicate an optional substitute

Description:

This command CATalogues the contents of a specified diskette to a given stream.

Examples:

10 CAT %0 catalogues drive 0.

10 FORMAT % #4; "T", 1200, 79

20 CAT % #4, 0 would send the CAT of drive 0 to the printer.

Command Name: CAT

Command Syntax:

CAT % (# <STREAM NUMBER> ,) <DRIVE NUMBER> (;B)
(;F)

Command Options:

STREAM NUMBER – This is a number of a stream that the contents of the CAT listing is to be sent to. Its default is the Screen.

DRIVE NUMBER – This is a number 0 to 3 that represents the drive which you want a CAT listing from.

B – This is the Brief option that lists only the filenames.

F – This is the Full option that lists additional data on the files.

Command Name: LOAD

Command Syntax:

LOAD %<DRIVE_NO ;><"FILENAME">(<CODE (LOAD ADDRESS(,LENGTH))>)
or (<DATA ARRAY NAME (\$)>)
or (<SCREEN\$>)

Command Options:

DRIVE_NO – is a number between 0 and 3 and is the drive from which you wish to load the program.

FILENAME – This is a string or a variable that is the name of the file that you wish to Load.

LOAD ADDRESS – is the address that you can optionally specify with the CODE option. This defaults to the start address specified when the file was saved.

LENGTH – is the length that you can optionally specify after the load address. The default is the whole file.

ARRAY NAME – is the name of a pre-defined array that you wish data to be loaded into.

Description:

This command loads from a file on a diskette, into a specified area of the Spectrum, in a specified format.

As with Sinclair BASIC, the additional CODE parameters may or may not be specified. If no parameters are specified, then those stored on the disc when the file was saved are used. (Refer to your Spectrum User's Manual).

An Image that has been saved on the diskette from the Interrupt button can be loaded from BASIC and will auto run.

If the file had been saved to the diskette, using the LINE option, (see save), it will auto run when it is loaded.

Examples:

LOAD %0;"MYFILE"	for a BASIC file, or
LOAD %0;"MYFILE" CODE	for a code file, or
LOAD %0;"MYFILE" DATA A()	for a data array, or
LOAD %0;"MYFILE" SCREEN\$	for a screen.

Command Syntax:

MERGE %DRIVE__NO;"FILENAME"

Command Options:

DRIVE__NO – Is a number between 0 and 3 and is the drive from which you wish to merge the program.

FILENAME – This is the name of the file that you wish to be merged with the program currently in the Spectrum.

Description:

This command loads a Basic file from the diskette and interleaves it with a program that is currently in RAM. If a line number already exists, then this line is overwritten with the line stored on disc.

Examples:

```
10 LOAD %0;"FILE1":REM LOAD THE ORIGINAL FILE
20 MERGE %0;"OVERLAY1":REM ADD NEW OVERLAY
```

Command Name: SAVE

Command Syntax:

SAVE % (DRIVE__NO;)<"FILENAME">(LINE number)
or (<DATA ARRAY NAME (\$)>)
or (<CODE START ADDRESS,
LENGTH>)
or (<SCREEN\$>)

Command Options:

DRIVE__NO – is a number between 0 and 3 that is the drive on which you wish to save the program.

FILENAME – This is a string or a variable that is the name of the file that you wish to Save.

START ADDRESS – is the address that you can optionally specify with the CODE option.

LENGTH – is the length that you can optionally specify after the start address.

ARRAY NAME – is the name of a pre-defined array that you wish data to be saved from.

Description:

This command saves a section of memory in a file on a drive. The type of file is dependent on the type of Save requested.

Examples:

SAVE %1;"FILENAME"	Saves a BASIC program on Drive 1.
SAVE %1;"FILENAME" DATA B\$()	Saves B\$ on Drive 1.
SAVE %1;"FILENAME" CODE 16384,8192	Saves a block of RAM on Drive 1.

Command Name: OPEN#

Command Syntax:

```
OPEN ###STREAM__NUMBER;DRIVE_NO;"FILENAME",<"R">,<"T">
                                     <"W">,<"R"><LEN>
                                     <"A">,
```

Command Options:

STREAM__NUMBER – This is the number of the stream that the file is attached to.

DRIVE_NO – Is a number between 0 and 3 on which the file resides.

FILENAME – This is a string of variable that contains the name of the file.

Operation –> R=Read W=Write A=Append

Type of file –> T=Text R=Fixed Length record.

LEN is the length of the records that is specified with the Fixed Length Record Option.

Description:

This command sets up a stream. One end of the stream points at a file that is OPENED for a particular type of operation, and the other end of the file is available to INPUT from and PRINT to, in the case of sequential files, or to OUT to or IN from in the case of Fixed Length Record Files .

(See Access Operations on a File).

Example:

```
10 OPEN ###4;0;"FRED","A","T"
20 PRINT #4;"This is the Swift Disc"
30 CLOSE ###4
```

This will take stream number 4 and connect to it the file FRED on drive 0. Therefore, any data that is pushed into stream 4 will actually go into the file FRED on the diskette. Line 20 PRINTS data onto the file FRED, and line 30 closes the stream and closes the file on the diskette.

To read back from the file, OPEN the file for read (line 40)

```
40 OPEN ###4;0;"FRED","R","T"
50 INPUT ###4;A$
60 PRINT A$
```

and then read the data back from it by INPUTting from that stream (line 50) which will obtain data from the file as opposed to the keyboard. This can then be printed onto the screen.

OPEN ###4;0;"FRED","R","T"	Sequential file open for read.
OPEN ###4;0;"FIXED","W","R",10	Record file, each record of length 10, open for write.
OPEN ###4;0;N\$,"A","T"	Text file open for append.

There is a limit of 4 files open at any one time, and all open files must reside on the same drive.

Command Name: CLOSE

Command Syntax:

CLOSE ### STREAM__NO

Command Options:

STREAM__NO – is a stream that has already been OPENed.

Description:

If data is being streamed to a file then you may notice that the disc interface waits until it has 256 bytes of data before it puts it into the file. This data is actually sitting in the stream waiting to go into the file. However, when you have finished you will want to empty all the data that was in the pipe onto the diskette. The close command does this. The close command either closes the file that was opened for write or append by purging the stream to the file, or by closing a stream that was open for read by clearing down all the internal pointers.

Note:

1. Do not use the Sinclair BASIC standard CLOSE # on a disc channel.
2. If the channel is not open this will cause a fatal crash due to a bug in the Sinclair PROM.

Example:

10 CLOSE ###4 Empty the stream and close the file.

Command Name: ERASE

Command Syntax:

ERASE %DRIVE__NO;FILENAME

Command Options:

DRIVE__NO – This is the drive number that the interface will erase the file from. This is a number between 0 and 3.

FILENAME – This is any legal filename on the diskette.

Description:

Erase, removes a file from the diskette, unless the protect option has been invoked from the button. Once a file has been erased, it cannot be unerased

Example:

10 SAVE %0;"NEWFILE"
20 ERASE %0;"OLDFILE"

OLDFILE is erased from the diskette.

Command Name: CLEAR

Command Syntax: CLEAR %

Command Options: None

Description:

This command resets all internal variables that are used by the disc interface. It clears the 'on error goto line number' if this has been specified, and also the %lower variable.

Example:

CLEAR % Clear down internal variables, only if files are not open. Reset %err.

Command Name: FORMAT

Command Syntax:

FORMAT %DRIVE__NO;"VOL__NAME",HEADERS,TRACKS

Command Options:

DRIVE__NO – This is the number of the drive that the interface will find the diskette to be formatted.

VOL__NAME – This is the volume name that is displayed in the CAT listing.

HEADERS – This is maximum number of files that can be stored on the diskette. It is to be between 13 (for games players) and 80. The more free headers you have, the less free blocks on the diskette. This defaults to 32.

TRACKS – This is the number of tracks on the Disc Drive. It is either 40 or 80. This defaults to 80 and only need be used if you are using NON Sixword supplied disc drives.

Description:

This command formats a diskette. This need only be done once.

Example:

10 FORMAT %0;"MYDISK" Format a diskette using defaults.
20 FORMAT %0;"40TRK",32,40 Format 40 track and 32 headers.

File Types:

There are two types of file. They are: –

- (1) Sequential files, (Text files in the open command). These are files that are used to store text in an unformatted mode. Inputting data from this type of file will return the data up until the next ENTER character. Printing to this type of file will store the data followed by an ENTER. You will search through all the elements to arrive at the element that you require. The commands that are designed to operate on this type of file are PRINT#, INPUT#, INKEY#.
- (2) Fixed length record files. (Record files in the open command). This type of file is accessed by an index number. The index number is used by the interface to skip past the records of no interest and present the record that is required.
The length of each record is set at file creation time and cannot be changed later. All records that are read from the file are of this fixed length.
Outputting data to a record will overwrite the contents that exist at that location. The commands that are designed to operate on this type of file are IN% or OUT%.

Access Operations on a file.

There are three ways in which to access a file. The first is to write to the beginning of a file. The second is to append new output to the end of an old file, and the third is to read data from a file.

A file therefore may be opened in one of three ways.

- (1) Open for read. This means that data may only be read from this type of file. When an INPUT#Stream__no is executed, data is read from the file up until the next ENTER character.
- (2) Open for Write. This is the way a file is created. In this case, data is written from the start of the file until the file is closed. A file that is open for write will overwrite any old data in that file from the beginning of the file. A file that is open for write is also open for read, but reading and writing should be used with care.
If an old file is opened for write and the new length is less than the old length, the file maintains the old length.
- (3) Open for append. Data is appended to the end of the file. Data files may be modified and extended using this operation. A file that is open for append is also open for read.
With fixed length record files, open for append has no meaning, as open for write automatically appends to the file.

Command Name: SEQUENTIAL FILE ACCESS

Command Syntax:

```
PRINT #STREAM__NUMBER;STRING  
INPUT #STREAM__NUMBER;VARIABLE  
VARIABLE = INKEY $ # STREAM__NUMBER
```

Command Options:

STREAM__NUMBER - This is the number of a stream (0 to 15) that you have opened (or in the case of the printer port, Formatted).

STRING - This is a string or a variable that is to be sent to the stream.

VARIABLE - This is a variable that the data that is read in from the stream is put into.

Description:

The PRINT command outputs data to a sequential file. The data outputted can either be an absolute string or a string variable. The INPUT and INKEY commands read in data from a stream. In the case of INPUT, up to the next enter character and in the case of INKEY, the next character.

Example:

```
10 PRINT #4;"This is the Swift Disc"  
20 PRINT #4;n$  
30 INPUT #4;a$  
40 LET A$=INKEY$ #4
```

Record Handling

The Swift Disc interface allows you to use fixed length records. These are treated in a different way to sequential files. The data is accessed via a record number i.e. an index into the file. For example, if you had a record length of say 6 and you wanted the 5th record, the interface multiplies the record length by the required record and returns the next record of the specified size from that point in the file. In the case above, 6 characters would be returned from the 30th character in the file. The commands IN% and OUT% are used to manipulate data with fixed length record files.

Command Name: IN%

Command Syntax:

IN %#<STREAM__NUMBER,ARRAY__NAME,RECORD__NO>

Command Options:

STREAM__NUMBER - is a stream that has been set up to a file.

ARRAY__NAME - is the name of the array in which you request that the data be put.

RECORD__NO - is the position index in the file where you want the data to be read from.

Description:

This command allows you to read a fixed length record from a file and store it in an array. The array element must be large enough to support the record that is being read in, otherwise it will be truncated.

Example:

```
10 DIM a(256)
20 IN %#4,A(0),3
```

This would read in record 3 from the stream and put it in array A.

Command Name: OUT %

Command Syntax:

OUT %#<STREAM__NUMBER,ARRAY__NAME,RECORD__NO>

Command Options:

STREAM__NUMBER - is a stream that has been set up to a file.

ARRAY__NAME - is the name of the array in which you request that data be read from.

RECORD__NO - is the position index in the file where you want the data to be put.

Description:

The OUT command writes data to a fixed length record file. The position in the file where the data is written is specified by the record number. The array is the place where the record is read from, and it must be the same size or larger than the record length.

If the record size is 5 bytes long, which is the internal size of a number in BASIC, then numbers and characters can be mixed within a record file.

Example:

```
10 OUT %#4,D(0),3
```

Would put array D(0) into record 3 of the file.

Command Name: %EOF

Command Syntax:

IF %EOF THEN

Command Options: None

Description:

A pseudo variable that is called %EOF (End of File) is defined internally by the interface and can be used by BASIC programs to determine if the last read was at end of file. This is available to both Fixed length record files and sequential files.

Example:

This expression must be placed after every read or write and is used as follows:

```
10 INPUT #4,A$
20 IF%eof THEN GOTO 50
30 GOTO 10
```

Command Name: %ERR

Command Syntax:

IF %ERR THEN GOTO LINE NUMBER
LET A = %ERR

Command Options: None

Description:

This command allows the programmer to set up an error path. When a disc error occurs, then the program flow can be directed to a common error handling routine. Within this error handler, the error line number (using %line) can be obtained. Also the error number can be obtained and thus the program can decode the source of the error and continue if required.

Example:

```
10 IF %ERR THEN GOTO 1000: REM SETS UP A GLOBAL ERROR HANDLER
20 CAT %4:REM CAUSE AN ERROR TO TEST IT

1000 REM DISC ERROR HANDLER
1010 LET E=%ERR:REM GET THE ERROR TYPE
1020 LET L=%LINE:REM GET THE LINE NUMBER WHERE THE
ERROR OCCURRED
1030 PRINT "ERROR TYPE";E;"ON LINE";L
1040 GOTO L:REM GO BACK TO THE LINE AND HAVE ANOTHER GO.
```

Command Name: %LINE

Command Syntax:

LET A = %LINE

Command Options: None

Description:

This command allows you to obtain the LINE number on which the last error occurred.

Example:

```
300 LET A=%LINE:REM GET THE LINE NUMBER WHERE THE ERROR  
OCCURRED
```

Command Name: %LOWER

Command Syntax:

LET %LOWER

Command Options: None

Description:

This command allows the upper case conversion within the interface to be turned off. Therefore file names that are upper case and lower case are allowed, and are treated as different files.

Note: From the button, lower case files cannot be moved or loaded back and return with the message 'file not found'. This lower flag is cleared by CLEAR %.

Example:

```
10 LET %lower
```

Command Name: %DATE

Command Syntax:

LET %DATE = D\$

Command Options: None

Description:

This command allows the change of the date held within the interface.

Example:

```
10 LET %DATE = D$:REM SET THE SYSTEM DATE.
```

Command Name: Printer Port FORMAT

Command Syntax:

```
FORMAT %#STREAM__NO;"T",BAUDRATE,CHARS__PER__LINE  
"B"
```

Command Options:

T - is for a Text stream, which is used to interpret tokens used by the Spectrum, (used for listings).

B - for binary which will let all characters pass without interpretation.

BAUDRATE - is either 50, 110, 300, 600, 1200, 2400, 4800, 9600, 19200.

Chars__per__line - is the number of characters on the line.

Description:

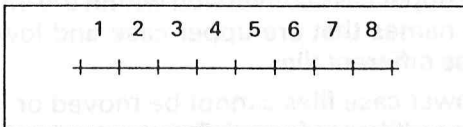
The printer port is suitable for any printer with a standard RS-232 serial interface.

The port is configured using the Format command.

The character format is 8 bits, two stop bits and no parity.

The pin allocation is as follows for the connector on the left hand side of the box.

The format command will allow you to format channel 3, which is the default printer output. Thus printer commands LPRINT, LLIST and LIST will operate on the printer directly.



1 not used

2 not used

3 not used

4 ground

5 DSR output

6 TX data output

7 DTR input

8 RX data input

An interfacing cable from the edge connector to a 25 way "D" type connector can be purchased from Sixword Ltd.

Example:

```
10 FORMAT %#4;"T",1200,79
```

The printer port is formatted as 1200 baud, 79 characters per line and as a text channel.

Data can then be listed to the port by

```
LIST #4
```

DO's and DONT's

Never turn on or off with a diskette in a drive.

Always treat the diskette with care, they are not indestructible.

Always back up to more than one diskette. Just in case ...

Always use branded 3.5" diskettes.

Avoid having the interface cable running alongside the T.V.

Do not leave diskettes next to the T.V. as data can be corrupted.

Do not poke things into the drive, other than diskettes.

Please do not try to take it apart. There are no user serviceable parts inside.

A file named RUN will autorun, if RUN is typed and there is no program in the Spectrum to run.

Errors and Warnings

Below are a list of errors and warnings that can be produced by the Interface:

DISC PROTECTED

The disc drive is unable to get data onto the diskette as the write protect shutter is not covering the hole on the diskette.

FILE ALREADY EXISTS

The file name that you have chosen already exists on the diskette. Either erase the file on the diskette or choose another name for the file that you wish to save.

BAD DISC BLOCK

This means that there was a checksum error when reading data from the diskette. If this persists then the file is then lost. This error is usually caused by mishandling the diskette.

Solution - reformat the diskette.

NO SUCH FILE

The file that was specified does not exist on the diskette. This is often caused by a typing error.

Solution - CAT the diskette and obtain the correct filename.

FILE PROTECTED

The file that you are trying to erase has been KEEPed and cannot be deleted.

Solution - Unkeep the file.

DISC LOST DATA

The interface has lost data during data transfer to or from the diskette. If this persists then the disc interface is faulty.

TRACK 0 FAILURE

This means that either the interface or the drive is faulty.

RECORD NOT FOUND

The interface thinks that the diskette is not formatted as the filing information on the diskette is not present.

Solution - Power OFF and ON. If persistent, reformat the diskette.

SYS ERR 1

A serious irrecoverable error has occurred within the interface normally associated with a loss of free memory.

SYS ERR 2

When the interface tried to access side 2 of the diskette, it accessed side 1. Can be caused by a loose drive cable.

INVALID DISC NUMBER

The disc number specified is greater than 3.

INSUFFICIENT SPACE

There is not enough free space on the diskette for the save.

Solution - remove files or use another diskette.

INVALID FILENAME

The filename specified was either greater than 10 characters or contained illegal characters.

Solution - respecify the file name.

SYNTAX ERROR

An error was made in the instruction that was issued to the interface.

DRIVE NOT PRESENT

A reference to a non-existent drive was made.

INVALID MDB

The master disc block has become invalid. Can be caused by turning the drive off with a diskette in place.

Solution - reformat the diskette.

TOO MANY OPEN FILES

A maximum of 4 open files at a time is allowed.

Appendix 1

NOTES

The CAT List

Catalogue of Drive: GAMES

INVADERS	192	PI	22-07-87
MYFILE	192	PT	23-07-87
SWIFT.EXE	192	I	10-09-87
TEMP.TAB	20	T	11-09-87
BAS.1	5	B	17-04-87

Total of 5 files

34 free headers out of 39

Free Disc Blocks 1740

At the top of the CAT listing is the Volume label given at Format time (GAMES). In the left hand column is the filename, followed by its size (in 256 byte blocks), followed by its type and lastly the date when it was saved. The filetypes are indicated below:

M = Memory Dump

I = Image file that was saved under the button.

P = The protect bit invoked by the Keep command.

T = Text file, sequential access.

D = Data file that was saved from basic.

S = Screen file.

R = Fixed length record file.

B = Basic program.

E = Emulator file.

NOTES

Appendix 1

The CAT List

Category of the Volume

INVADEHS	110	PL	2-04-87
MYRLE	111	MT	20-01-87
SWPTXSE	102	I	20-03-87
TEMPYTAO	20	T	11-03-87
BAP 1	8	B	17-04-87

Final of 8 files
 24 files deleted in 1/88
 Files Dir: Block 1, 2, 3

At the top of the CAT listed in the Volume label (even if it is not the
 (GAMER) in a label found normal in the file is followed by its size in
 256 bytes and followed by the type and finally the date that it was saved.
 The file type is indicated below:

- M - Memory dump
- I - Image file that was saved under the system
- P - The protocol file invoked by the kernel command
- T - Text file, sequential access
- D - Data file that was saved from disk
- S - Screen file
- H - Fixed length record file
- B - Basic program
- F - Executable file

